ARMY RESEARCH LABORATORY

ROStoJAUSBridge Manual

by Laurel Sadler, Chirag Rao, John Rogers, and Hung Nguyen

ARL-MR-0812 **March 2012**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-MR-0812 **March 2012**

ROStoJAUSBridge Manual

Laurel Sadler, Chirag Rao, John Rogers, and Hung Nguyen
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
March 2012	Final	

4. TITLE AND SUBTITLE
ROStoJAUSBridge Manual

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Laurel Sadler, Chirag Rao, John Rogers, and Hung Nguyen

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-CII-B
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-0812

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The ROStoJAUSBridge is a software program implemented at the U.S. Army Research Laboratory (ARL) to leverage the image processing and intelligent mobility behaviors capabilities from the robot operating system (ROS) to the small unmanned ground vehicle (SUG)V. This software program allows for interaction between programs running on ROS and the iRobot SUGV. It converts ROS messages to messages adhering to the Joint Architecture for Unmanned Systems (JAUS) architecture and vice versa. The supported set of messages is limited to query and receive the velocity state of the SUGV which will be used with the inertial measurement unit (IMU) data to compute the local pose; and to send the velocity commands from ROS' programs to the SUGV

15. SUBJECT TERMS
Robot operating system, software packages

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	54	Laurel Sadler
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (*Include area code*) (301) 394-1221

Contents

List of Figures iv

List of Tables iv

1. Introduction 1

2. ROStoJAUSBridge Overview 2

3. ROStoJAUSBridge Layout 4

3.1 Using ROStoJAUSBridge 6

3.1.1 Controlling the SUGV's Tread and Flipper Motion 6

3.1.2 Changing the Robot's Internet Protocol (IP) Address 7

3.2 Future Tasks and Improvements 7

4. References 8

Appendix. ROStoJAUSBridge API 9

List of Symbols, Abbreviations, and Acronyms 46

Distribution List 47

List of Figures

Figure 1. Message flow between the SUGV and ROS via the ROStoJAUSBridge. 2

Figure 2. (a) This depicts the command message flow from ROS to the SUGV to drive the SUGV and (b) this depicts the command message flow of the odometry data from the SUGV to ROS. 3

Figure 3. This depicts data flow between the ROStoJAUSBridge and various ROS packages. 4

List of Tables

Table 1. JAUS messages supported by ROStoJAUSBridge.....6

1. Introduction

The Asset Control & Behavior Branch of the U.S. Army Research Laboratory (ARL) is currently conducting research in the area of autonomy for small unmanned ground vehicles (SUGVs). The SUGV can be deployed with the Soldier for improved situational awareness and to perform tasks that are considered dangerous or life-threatening. The SUGV can provide information about the area it is autonomously exploring to the Soldier, who remains in a safe environment. The SUGV is controlled by and communicates with other systems via the Joint Architecture for Unmanned Systems (JAUS) messaging architecture (*1*).

Baseline autonomous capabilities, including mapping and navigation, have been developed for the SUGV platforms, making use of the open-source Robot Operating System (ROS) software from Willow Garage, Inc. (*5*) ROS provides libraries and tools for hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. To achieve the autonomous exploration and navigation components of the baseline system, the software integration on the SUGV consisted of: (1) various ROS packages running on the payload computer to provide exploration, mapping, and navigation, as well as two hardware drivers for sensors; (2) the JAUS communications architecture software running on the SUGV internal processor to provide access to low-level control; and (3) a software bridge between JAUS and ROS to allow transference of velocity commands, odometry, and other data between them.

For our specific application, we used the ROS low-level device control for the Hokuyo UTM-30LX Laser Radar (Ladar) and the Microstrain 3DM-GX inertial measurement unit (IMU), in addition to multiple ROS **s**oftware packages for mapping, and navigation for local and global mapping. The ROS mapping software package GMapping is an implementation of Simultaneous Localization and Mapping (SLAM) using a Rao-Blackwellized particle filter to build occupancy grid maps from laser data (*2*). Since this algorithm is sensitive to the quality of odometry inputs that related successive scans, it was necessary to implement the ROS node (odom_imu_to_tf), which fuses the SUGV raw odometry data with the IMU data by maintaining an orientation estimate from the integrated IMU angular velocity measurement instead of the estimated turning velocity from the SUGV. The ROS navigation software package (move_base) provides a 2-D navigation stack that requires input from sensor streams, odometry, and a goal position provided by the exploration package; it also outputs safe velocity commands. The ROS navigation package consists of both a global and local planner. The global planner operates on a costmap provided by Gmapping to find a minimum cost plan between two points on a grid using the Dijkstra's algorithm. The local planner implemented is based on the Dynamic Window Approach (DWA) (*3, 4*). This planner makes use of the global plan and the associated costmap

as inputs and provides the velocity commands as outputs. The command/data flow between each of the individual ROS packages and the ROStoJAUSBridge is shown in figure 3.

2. ROStoJAUSBridge Overview

The ROStoJAUSBridge is a software program running on the SUGV payload that allows for interaction between the ROS packages also running on the payload computer and the JAUS messaging architecture/mobility controller on the SUGV platform computer. It was developed to enable communication between the ROS device controllers and autonomy packages, with the JAUS messaging architecture controlling the SUGV by providing callbacks to both systems that translate corresponding messages between them, as shown in figure 1.

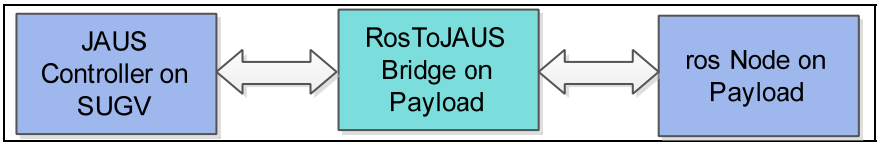


Figure 1. Message flow between the SUGV and ROS via the ROStoJAUSBridge.

In this application, the ROStoJAUSBridge connects to the JAUS controller on the SUGV and requests status commands parsing the SUGV odometry data. The bridge then publishes this data, which includes x and y position, heading information, and translational and rotational velocities, to the ROS packages. The ROStoJAUSBridge also subscribes to mobility commands being published by the ROS navigation package to autonomously drive the SUGV by sending a “set wrench effort” command to JAUS. This is shown in figures 2a and 2b, which further break down the block diagram shown in figure 1.

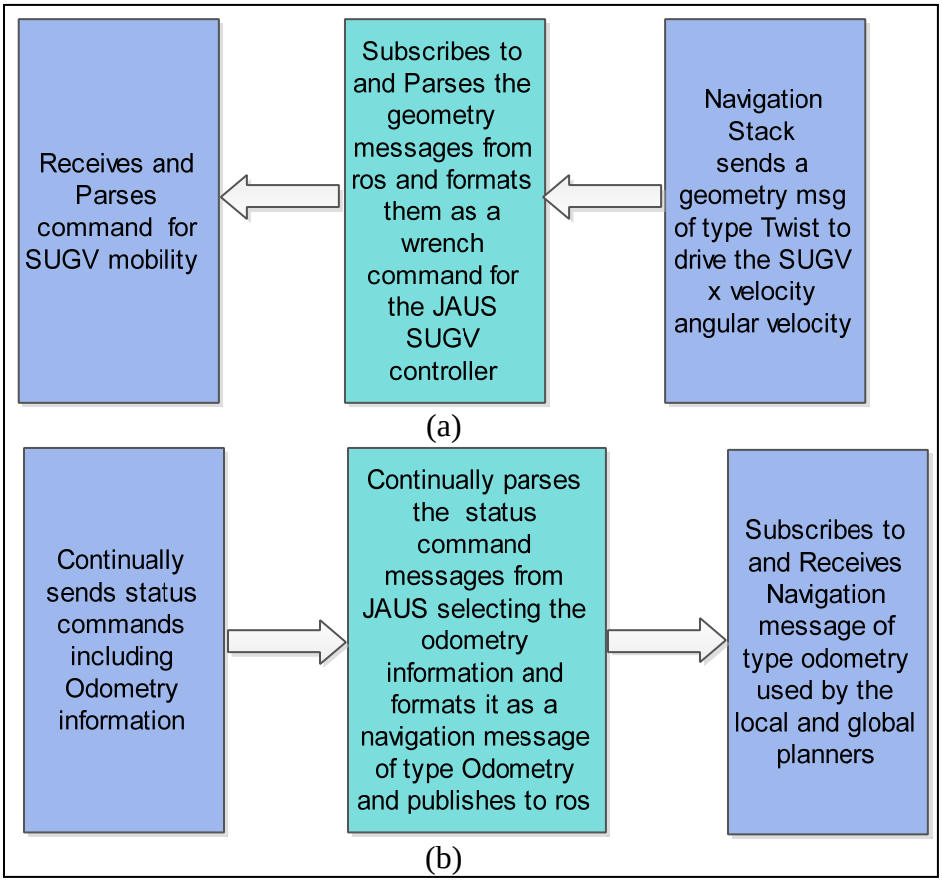


Figure 2. (a) This depicts the command message flow from ROS to the SUGV to drive the SUGV and (b) this depicts the command message flow of the odometry data from the SUGV to ROS.

As depicted in figure 3, the primary functions of the ROStoJAUSBridge are to (1) receive the cmd_vel command from the ROS navigation package or move_base and (2) send the odometry information to odom_imu_to_tf to be processed for improved IMU data, which is then sent to the navigation package, move_base, and to the slam package to be used for mapping.

What is not shown in figure 3 are the inputs and outputs from ROStoJAUSbridge to the JAUS interface on the SUGV as previously discussed. Briefly stated, the cmd_vel command from ROS is formatted for JAUS and sent to the SUGV, and the odometry data received from the SUGV is parsed and reformatted as a ROS odometry message and sent to the odom_imu_to_tf package.

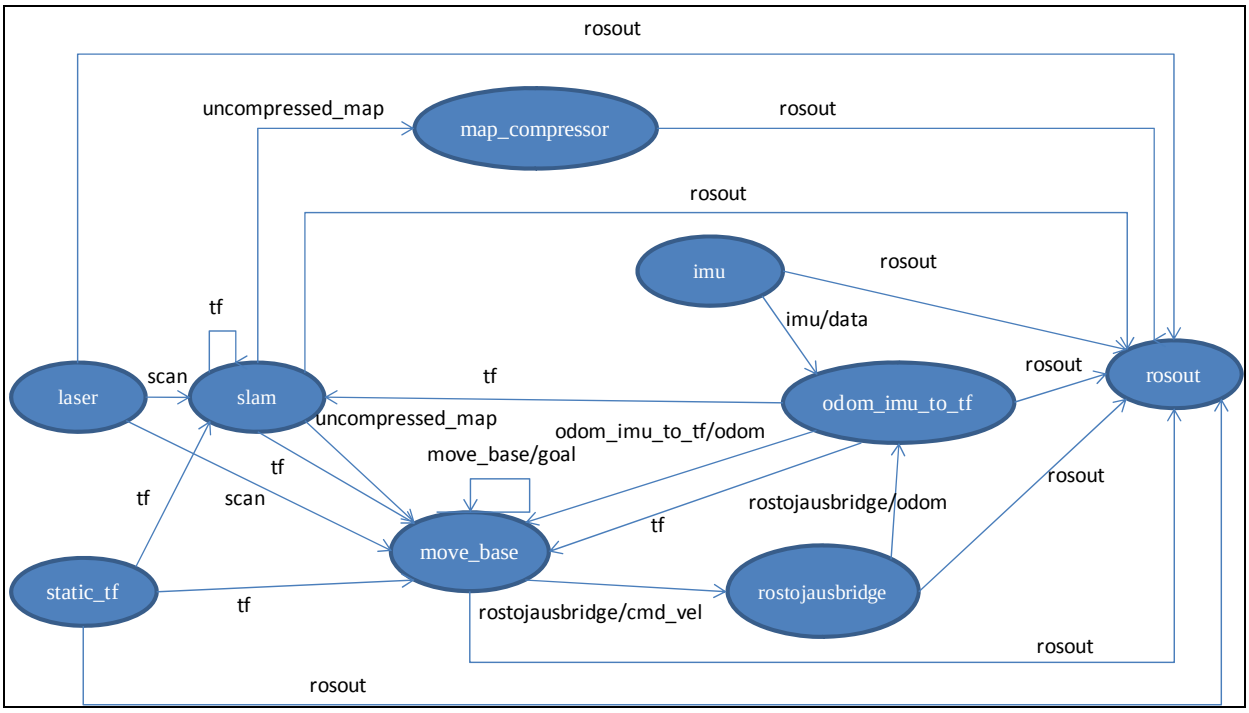


Figure 3. This depicts data flow between the ROStoJAUSBridge and various ROS packages.

In addition, the mobility command published by the ROS navigation node also provides information for controlling the flippers on the SUGV. The ROStoJAUSBridge parses the flipper control information and also allows for flipper control on the SUGV by sending a "set joint effort" command. This is an important capability for maneuvering the SUGV when considering stair-climbing and rough terrain for SUGV navigation. It may also be necessary to return the flippers to a more compact position when maneuvering the SUGV through narrow passages or turning around in tight spaces.

The ROStoJAUSBridge is currently implemented in the C++ programming language using standard C++ libraries, as well as the "Diamondback" distribution of ROS libraries on an Ubuntu 10.10 Linux system. Currently, the ROStoJAUSBridge is designed to communicate only with the SUGV system. However, it can be easily adapted to other UGV platforms using the JAUS architecture. More detailed class documentation for the ROStoJAUSBridge API can be found in the appendix.

3. ROStoJAUSBridge Layout

The program is comprised of two source code files: ROStoJAUSBridge.cpp and JAUSmessages.cpp, each with a corresponding header file. ROStoJAUSBridge.cpp is the executable software package that handles User Datagram Protocol (UDP) socket formation and

publishing/subscribing to ROS topics. ROStoJAUSBridge.cpp uses the JAUS message classes defined in JAUSmessages.cpp to send and receive messages to and from the robot.

Within ROStoJAUSBridge.cpp, the ROStoJAUSBridge class is responsible for connecting to and maintaining the UDP socket over which messages will be exchanged between the ROStoJAUSBridge and the SUGV. The ROStoJAUSBridge class also defines the node handler publisher and subscriber classes that allow for information to pass between the ROStoJAUSBridge program and the ROS navigation packages. ROStoJAUSBridge.cpp also defines the local_pose_calculator class, which uses a dead reckoning approach to determine the robot's local position and orientation. The local_pose_calulator class uses input data from the odometry data received from the SUGV in combination with the improved IMU data received from ROS. The ROStoJAUSBridge then returns the robot's local position and orientation information to the ROS Navigation Stack to be used for Global and Local Planning. Figure 3 depicts the interactions between the various ROS nodes and the ROSToJAUSBridge to accomplish this and other tasks. To understand in greater detail how local_pose_calculator calculates the robot's position, please refer to Thrun, Burgard, and Fox's *Probabilistic Robotics* (2005, p. 125-127).

The JAUSmessages.cpp source code defines classes that format information into a JAUS message and parse messages received from the SUGV. The JAUSmessages header file defines global functions that are used extensively in ROStoJAUSBridge.

Table 1 gives an overview of the messages JAUSmessages.cpp supports at this time. Each message has a distinct 2-byte command code (shown in column 2 of table 1 in hexadecimal numbers), and each message is designed for one-way communication only—e.g., the Query_Velocity_State message is only sent from the Operator Control Unit (OCU) to the SUGV and will not be sent the other way. The messages highlighted in blue are user-defined messages, meaning these messages are specific to the SUGV platform. User-defined messages differ from general JAUS messages, in that the command codes' most significant quartet will be "F." For more detailed information on how JAUS message packets are structured, and how they can be parsed, please refer to the SUGV Interface Design Description (2009, section 4.4) for user-defined messages and the JAUS Reference Architecture Specification (2004, vol. II, part 3) for general JAUS messages.

Table 1. JAUS messages supported by ROStoJAUSBridge.

Message	Command Code	Message Destination	Message Source
Request_Component_Control	0x000D	OCU	SUGV
Query_Wrench_Effort	0x2405	SUGV	OCU
Query_Platform_Operational_Data	0x2401	SUGV	OCU
Set_Joint_Efforts	0x0601	SUGV	OCU
Set_Wrench_Effort	0x0405	OCU	SUGV
Report_Global_Pose	0x4402	OCU	SUGV
Report_Velocity_State	0x4404	OCU	SUGV
Report_Camera_Control	0xF008	OCU	SUGV
Report_AntiCollision	0xF00D	OCU	SUGV
Report_Motor_Status	0xF00B	OCU	SUGV
Report_Battery_Status	0xF004	OCU	SUGV
Report_Platform_Operational_Data	0x4401	OCU	SUGV
Report_Illuminator_Intensity	0xF00A	OCU	SUGV
Report_Power_Control	0xF002	OCU	SUGV
Report_Selected_Camera	0x4804	OCU	SUGV
Report_Latch_Control	0xF00E	OCU	SUGV
Report_SUGV_Telemetry	0xF00C	OCU	SUGV
Report_Calibrate_Information	0xF007	OCU	SUGV
Report_Version_Info	0xF010	OCU	SUGV
Report_Payload_Identification	0xF009	OCU	SUGV
Query_Velocity_State	0x2404	SUGV	OCU
Query_Global_Pose	0x2402	SUGV	OCU

3.1 Using ROStoJAUSBridge

3.1.1 Controlling the SUGV's Tread and Flipper Motion

The program currently subscribes and publishes to two topics, which are respectively "cmd_vel," and "odom." The cmd_vel topic broadcasts ROS messages of type geometry_msgs/Twist, and odom broadcasts nav_msgs/Odometry messages. To control the robot's linear velocity along the x-axis in a 3D Cartesian coordinate space, linear.x in the Twist message must be set to a floating point value between –3.0 and 3.0 m/s. To control the yaw rate about the z-axis, angular.z must be set to a floating point value between –3.0 and 3.0 radians per second. Finally, to control the robot's flipper motion, angular.y must be set to a value between –1.0 and 1.0 radian per second.

3.1.2 Changing the Robot's Internet Protocol (IP) Address

Prior to execution of the ROSto JAUSBridge, the IP address of the robot can be specified in a ROS launch file that includes the call to ROStoJAUSBridge. This is the only method by which the robot's IP address can be specified; otherwise, the default robot IP address will be "192.168.130.142." Currently, the robot's IP address cannot be passed to the program as a "main" argument at the point of execution. This may be a feature worth adding to ROStoJAUSBridge.

3.2 Future Tasks and Improvements

The fundamental objective that ROStoJAUBridge was to publish the SUGV's odometry information to ROS and subscribe to the mobility commands to autonomously drive the SUGV. As a result, only a few messages sent to and received from the SUGV were actually used or parsed. Also, not all the messages that can be sent to or received from the SUGV have been implemented in the program at this time. ROStoJAUSBridge will be expanded to support any JAUS messages and SUGV-specific messages as needed.

4. References

1. The JAUS 3.2 Reference Architecture Specification, http://www.jauswg.org/base.ine/refarch.html
2. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics* **2006**, *23* (1), 34–46.
3. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation* **1997**, *4* (1).
4. Gerkey, B. P.; Konolige, K. Planning and Control in Unstructured Terrain. *Proc. of the ICRA Workshop on Path Planning on Costmaps*, Pasadena, CA, 2008.
5. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A. ROS: An Open-source Robot Operating System. *in Open-source software workshop of the Int. Conf. on Robotics and Automation (ICRA)*, 2009.

Appendix. ROStoJAUSBridge API

ROStoJAUSBridge API

US Army Research Laboratory
Asset Control and Behavior Branch
Version 1.0
Date 2011-08-16

2

Table of Contents

Table of contents

ROStoJAUSBridge API

Version:
1.0

Author:
Army Research Laboratory - Asset Control and Behavior Branch

Date:
2011-08-16

Introduction

ROS to JAUS Bridge is a software program implemented by Laurel Sadler and Chirag Rao at ARL to leverage the images processing and intelligent mobility behaviors capabilities from ROS to SUGV. This software program allows for interaction between programs running on ROS and the iRobot SUGV. It converts ROS messages to messages adhering to the JAUS architecture and vice versa. The supported set of messages is limited to query and receive the velocity state of the SUGV which will be used with the IMU data to compute the local pose; and to send the velocity commands from ROS' programs to the SUGV.

Class Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

anti_collision....5
calibrate_info....7
local_pose_calculator....15
motor_stat....18
payload_ID....19
ROStoJAUSBridge....26
UDPmessage....29
anticollision_msg....5
battery_stat_msg....6
cam_ctrl_msg....7
component_control_msg....8
flipper_effort_msg....9
global_pose_msg....10
head_neck_effort_msg....10
illuminator_msg....11

jaus_header_msg....12
joint_positions_msg....13
latch_ctrl_msg....14
manipulator_effort_msg....16
motion_msg....17
motor_stat_msg....18
payload_ID_msg....19
platform_op_data_msg....20
power_ctrl_msg....21
query_global_pose_msg....22
query_platform_op_data_msg....23
query_velocity_msg....24
query_wrench_effort_msg....24
report_calibrate_device_msg....25
selected_camera_msg....28
SUGV_telem_msg....28
velocity_msg....33
version_info_msg....34
wrench_effort_msg....35

version_info....34

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

anti_collision 5
anticollision_msg 5
battery_stat_msg 6
calibrate_info 7
cam_ctrl_msg 7
component_control_msg 8
flipper_effort_msg 9
global_pose_msg 10
head_neck_effort_msg 10
illuminator_msg 11
jaus_header_msg 12
joint_positions_msg 13
latch_ctrl_msg 14
local_pose_calculator 15
manipulator_effort_msg 16
motion_msg 17
motor_stat 18
motor_stat_msg 18
payload_ID 19

payload_ID_msg 19
platform_op_data_msg 20
power_ctrl_msg 21
query_global_pose_msg 22
query_platform_op_data_msg 23
query_velocity_msg 24
query_wrench_effort_msg 24
report_calibrate_device_msg 25
ROStoJAUSBridge 26
selected_camera_msg 28
SUGV_telem_msg 28
UDPmessage 29
velocity_msg 33
version_info 34
version_info_msg 34
wrench_effort_msg 35

Class Documentation

anti_collision Struct Reference

Public Member Functions

1 **anti_collision** (unsigned char stat, unsigned short ID1, unsigned short ID2)

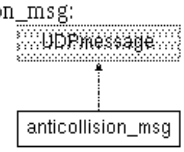
Public Attributes

2 unsigned char **status**
3 unsigned short **obj_ID1**
4 unsigned short **obj_ID2**

The documentation for this struct was generated from the following file:
5 JAUSmessages.h

anticollision_msg Class Reference

Inheritance diagram for anticollision_msg:



Public Member Functions

6 virtual size_t **size** () const
7 virtual unsigned char * **marshal** (unsigned char *out_array) const
8 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
9 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * anticollision_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t anticollision_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * anticollision_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

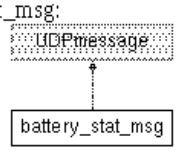
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

10 JAUSmessages.h
11 JAUSmessages.cpp

battery_stat_msg Class Reference

Inheritance diagram for battery_stat_msg:



Public Member Functions

12 virtual size_t **size** () const
13 virtual unsigned char * **marshal** (unsigned char *out_array) const
14 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
15 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * battery_stat_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t battery_stat_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * battery_stat_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

16 JAUSmessages.h
17 JAUSmessages.cpp

calibrate_info Struct Reference

Public Member Functions

18 **calibrate_info** (unsigned char ID, unsigned char stat)

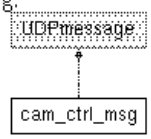
Public Attributes

19 unsigned char **device_ID**
20 unsigned char **status**

The documentation for this struct was generated from the following file:
21 JAUSmessages.h

cam_ctrl_msg Class Reference

Inheritance diagram for cam_ctrl_msg:



Public Member Functions

22 virtual size_t **size** () const
23 virtual unsigned char * **marshal** (unsigned char *out_array) const
24 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
25 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * cam_ctrl_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t cam_ctrl_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * cam_ctrl_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

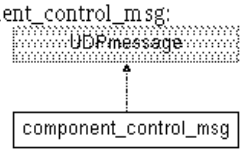
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

26 JAUSmessages.h
27 JAUSmessages.cpp

component_control_msg Class Reference

Inheritance diagram for component_control_msg:



Public Member Functions

28 virtual size_t **size** () const
29 virtual unsigned char * **marshal** (unsigned char *outp_array) const
30 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
31 unsigned short **CommandCode** () const

Member Function Documentation

unsigned char * component_control_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t component_control_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * component_control_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

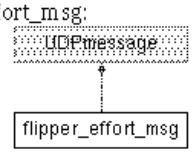
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

32 JAUSmessages.h
33 JAUSmessages.cpp

flipper_effort_msg Class Reference

Inheritance diagram for flipper_effort_msg:



Public Member Functions

34 **flipper_effort_msg** (float effort)
35 **flipper_effort_msg** (unsigned char eff_lo, unsigned char eff_hi)
36 virtual size_t **size** () const
37 virtual unsigned char * **marshal** (unsigned char *outp_array) const
38 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * flipper_effort_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t flipper_effort_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * flipper_effort_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

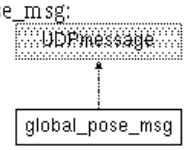
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

39 JAUSmessages.h
40 JAUSmessages.cpp

global_pose_msg Class Reference

Inheritance diagram for global_pose_msg:



Public Member Functions

41 virtual size_t **size** () const
42 virtual unsigned char * **marshal** (unsigned char *outp_array) const
43 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
44 const unsigned char * **unmarshal** (const unsigned char *inp_array, const **jaus_header_msg** &head)

Member Function Documentation

unsigned char * global_pose_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t global_pose_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * global_pose_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

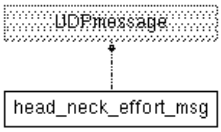
The documentation for this class was generated from the following files:

45 JAUSmessages.h
46 JAUSmessages.cpp

head_neck_effort_msg Class Reference

Inheritance diagram for head_neck_effort_msg:

10



Public Member Functions

47 virtual size_t **size** () const
48 virtual unsigned char * **marshal** (unsigned char *outp_array) const
49 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * head_neck_effort_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t head_neck_effort_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * head_neck_effort_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

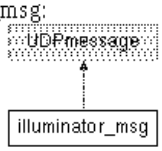
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

50 JAUSmessages.h
51 JAUSmessages.cpp

illuminator_msg Class Reference

Inheritance diagram for illuminator_msg:



Public Member Functions

52 virtual size_t **size** () const
53 virtual unsigned char * **marshal** (unsigned char *out_array) const
54 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
55 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * illuminator_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t illuminator_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * illuminator_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

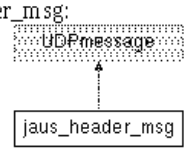
The documentation for this class was generated from the following files:

56 JAUSmessages.h
57 JAUSmessages.cpp

jaus_header_msg Class Reference

#include <JAUSmessages.h>

Inheritance diagram for jaus_header_msg:



Public Member Functions

58 **jaus_header_msg** (unsigned char dest_instance_id, unsigned char dest_component_id, unsigned short command_code, unsigned short message_body_length)
59 **jaus_header_msg** (unsigned char dest_instance_id, unsigned char dest_component_id, unsigned char dest_node_id, unsigned char dest_subsystem_id, unsigned char src_instance_id, unsigned char src_component_id, unsigned char src_node_id, unsigned char src_subsystem_id, unsigned short command_code, unsigned short message_body_length)
60 virtual size_t **size** () const
61 virtual unsigned char * **marshal** (unsigned char *outp_array) const
62 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
63 unsigned char **getSource_instance_ID** ()
64 unsigned char **getSource_component_ID** ()
65 unsigned short **CommandCode** () const
66 const unsigned char * **skip_msg** (const unsigned char *buff)

Detailed Description

A class that handles forming the JAUS header portion of JAUS messages. **jaus_header_msg** is included as a data member in all JAUS messages.

Member Function Documentation

unsigned char * jaus_header_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t jaus_header_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * jaus_header_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

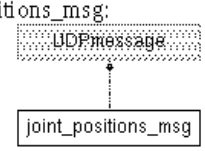
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

67 JAUSmessages.h
68 JAUSmessages.cpp

joint_positions_msg Class Reference

Inheritance diagram for joint_positions_msg:



Public Member Functions

69 virtual size_t **size** () const
70 virtual unsigned char * **marshal** (unsigned char *outp_array) const
71 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
72 const unsigned char * **unmarshal** (const unsigned char *inp_array, const **jaus_header_msg** &head)
73 unsigned int **getFlipper_position** ()

Member Function Documentation

unsigned char * joint_positions_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t joint_positions_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * joint_positions_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

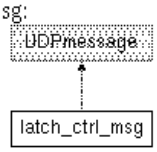
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

74 JAUSmessages.h
75 JAUSmessages.cpp

latch_ctrl_msg Class Reference

Inheritance diagram for latch_ctrl_msg:



Public Member Functions

76 virtual size_t **size** () const
77 virtual unsigned char * **marshal** (unsigned char *out_array) const
78 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
79 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * latch_ctrl_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t latch_ctrl_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * latch_ctrl_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

80 JAUSmessages.h
81 JAUSmessages.cpp

local_pose_calculator Class Reference

`#include <ROStoJAUSBridge.h>`

Public Member Functions

82 **local_pose_calculator** ()
83 float **getLoc_x** ()
84 float **getLoc_y** ()
85 float **getTheta** ()
86 float **getVel_x** ()
87 float **getVel_theta** ()
88 double **getSec** ()
89 float **getFlipper_position** ()
90 void **updatePose** (float vel_x, float vel_theta, double time)
91 void **updatePose** (float flipper)

Detailed Description

A class that calculates the robot's local pose. This class calculates the robot's local pose using velocity state information, including x-directional linear velocity and the yaw rate about the z-axis in a 3-dimensional Cartesian coordinate space.

Constructor & Destructor Documentation

local_pose_calculator::local_pose_calculator ()

Constructor for **local_pose_calculator** class. All pose variables are set to zero. isValid flag is set to false.

Member Function Documentation

float local_pose_calculator::getFlipper_position ()

Returns orientation of flipper (radians)

float local_pose_calculator::getLoc_x ()

Returns x coordinate (meters).

float local_pose_calculator::getLoc_y ()

Returns y coordinate (meters).

double local_pose_calculator::getSec ()

Returns timestamp (seconds).

float local_pose_calculator::getTheta ()

Returns local orientation with respect to x-axis (radians).

float local_pose_calculator::getVel_theta ()

Returns yaw rate about the z-axis (radians/sec).

float local_pose_calculator::getVel_x ()

Returns x-directional linear velocity (meters/sec).

void local_pose_calculator::updatePose (float *flipper*)

Updates the flipper position. flipper_position is assigned value flipper.

void local_pose_calculator::updatePose (float *vel_x*, float *vel_theta*, double *time*)

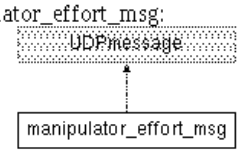
Updates local pose and timestamp. x,y coordinates and orientation are updated using the Velocity Motion Model ("Probabilistic Robotics," by Thrun, Burgard, and Fox, 2005, p.125-127)

The documentation for this class was generated from the following files:

92 ROStoJAUSBridge.h
93 ROStoJAUSBridge.cpp

manipulator_effort_msg Class Reference

Inheritance diagram for manipulator_effort_msg:



Public Member Functions

94 virtual size_t **size** () const
95 virtual unsigned char * **marshal** (unsigned char *outp_array) const
96 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * manipulator_effort_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t manipulator_effort_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * manipulator_effort_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

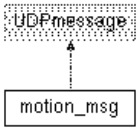
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

97 JAUSmessages.h
98 JAUSmessages.cpp

motion_msg Class Reference

Inheritance diagram for motion_msg:



Public Member Functions

99 **motion_msg** (float flipper_effort, float linx, float angz)
100 **motion_msg** (unsigned char flip_lo, unsigned char flip_hi, float linx, float angz)
101 virtual size_t **size** () const
102 virtual unsigned char * **marshal** (unsigned char *outp_array) const
103 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * motion_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t motion_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * motion_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

104 JAUSmessages.h
105 JAUSmessages.cpp

motor_stat Struct Reference

Public Member Functions

106 **motor_stat** (char ID, int temp, int greetings)

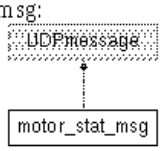
Public Attributes

107 char **motor_ID**
108 int **temperature**

The documentation for this struct was generated from the following file:
109 JAUSmessages.h

motor_stat_msg Class Reference

Inheritance diagram for motor_stat_msg:



Public Member Functions

110 virtual size_t **size** () const
111 virtual unsigned char * **marshal** (unsigned char *out_array) const
112 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

113 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * motor_stat_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t motor_stat_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * motor_stat_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

114 JAUSmessages.h
115 JAUSmessages.cpp

payload_ID Struct Reference

Public Member Functions

116 **payload_ID** (unsigned char bay, unsigned char id, unsigned char len, char *info)
117 **payload_ID** (const **payload_ID** &pay)
118 **payload_ID** & **operator=** (const **payload_ID** &pay)

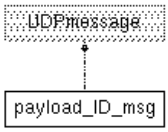
Public Attributes

119 unsigned char **payload_bay**
120 unsigned char **ID**
121 unsigned char **description_length**
122 char * **description_string**

The documentation for this struct was generated from the following file:
123 JAUSmessages.h

payload_ID_msg Class Reference

Inheritance diagram for payload_ID_msg:



Public Member Functions

124 virtual size_t **size** () const
125 virtual unsigned char * **marshal** (unsigned char *out_array) const
126 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
127 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * payload_ID_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t payload_ID_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * payload_ID_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

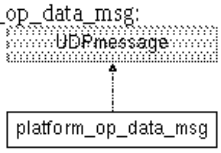
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

128 JAUSmessages.h
129 JAUSmessages.cpp

platform_op_data_msg Class Reference

Inheritance diagram for platform_op_data_msg:



Public Member Functions

130 virtual size_t **size** () const
131 virtual unsigned char * **marshal** (unsigned char *out_array) const
132 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
133 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * platform_op_data_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t platform_op_data_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * platform_op_data_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

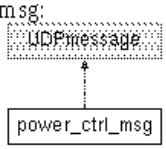
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

134 JAUSmessages.h
135 JAUSmessages.cpp

power_ctrl_msg Class Reference

Inheritance diagram for power_ctrl_msg:



Public Member Functions

136 virtual size_t **size** () const
137 virtual unsigned char * **marshal** (unsigned char *out_array) const
138 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
139 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * power_ctrl_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t power_ctrl_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * power_ctrl_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

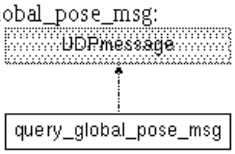
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

140 JAUSmessages.h
141 JAUSmessages.cpp

query_global_pose_msg Class Reference

Inheritance diagram for query_global_pose_msg:



Public Member Functions

142 **query_global_pose_msg** (unsigned char dest)
143 virtual unsigned char * **marshal** (unsigned char *out_array) const
144 virtual size_t **size** () const
145 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * query_global_pose_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t query_global_pose_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * query_global_pose_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

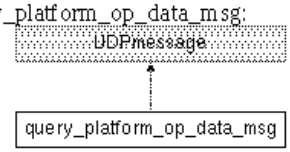
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

146 JAUSmessages.h
147 JAUSmessages.cpp

query_platform_op_data_msg Class Reference

Inheritance diagram for query_platform_op_data_msg:



Public Member Functions

148 **query_platform_op_data_msg** (unsigned char dest)
149 virtual size_t **size** () const
150 virtual unsigned char * **marshal** (unsigned char *outp_array) const
151 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * query_platform_op_data_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t query_platform_op_data_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * query_platform_op_data_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

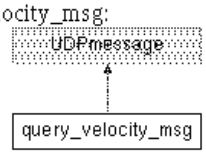
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

152 JAUSmessages.h
153 JAUSmessages.cpp

query_velocity_msg Class Reference

Inheritance diagram for query_velocity_msg:



Public Member Functions

154 **query_velocity_msg** (unsigned char dest)
155 virtual unsigned char * **marshal** (unsigned char *out_array) const
156 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
157 virtual size_t **size** () const

Member Function Documentation

unsigned char * query_velocity_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t query_velocity_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * query_velocity_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

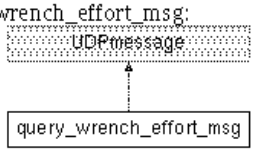
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

158 JAUSmessages.h
159 JAUSmessages.cpp

query_wrench_effort_msg Class Reference

Inheritance diagram for query_wrench_effort_msg:



Public Member Functions

160 **query_wrench_effort_msg** (unsigned char dest)
161 virtual size_t **size** () const
162 virtual unsigned char * **marshal** (unsigned char *outp_array) const
163 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * query_wrench_effort_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t query_wrench_effort_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * query_wrench_effort_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

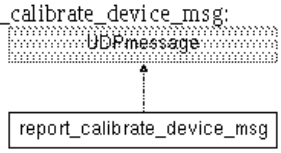
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

164 JAUSmessages.h
165 JAUSmessages.cpp

report_calibrate_device_msg Class Reference

Inheritance diagram for report_calibrate_device_msg:



Public Member Functions

166 virtual size_t **size** () const
167 virtual unsigned char * **marshal** (unsigned char *out_array) const
168 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
169 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * report_calibrate_device_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t report_calibrate_device_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * report_calibrate_device_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

170 JAUSmessages.h
171 JAUSmessages.cpp

ROStoJAUSBridge Class Reference

`#include <ROStoJAUSBridge.h>`

Public Member Functions

172 **ROStoJAUSBridge** ()
173 **~ROStoJAUSBridge** ()
174 void **timerCallback** (const ros::TimerEvent &te)
175 void **CmdVelCallback** (const geometry_msgs::TwistConstPtr &msg)
176 void **parse_msgs** (const unsigned char *buffer, size_t)

Protected Member Functions

177 void **onRobotTelemetry** ()
178 bool **ConnectToSUGV** ()

Protected Attributes

179 volatile float **vel_x**
180 volatile float **vel_y**
181 volatile float **vel_theta**
182 std::string **odom_frame**
183 std::string **base_frame**
184 std::string **robot_ip**
185 ros::NodeHandle **nh**
186 unsigned char **response_buffer** [RESP_BUFF_SIZE]
187 ros::Subscriber **mobcmd_sub**

188 ros::Publisher **odom_pub**
189 int **sd**
190 int **rc**
191 int **flags**
192 int **timeOut**
193 int **error**
194 struct sockaddr_in cliAddr remoteServAddr servAddr **remoteCliAddr**
195 unsigned int **remoteCliLen**
196 struct hostent * **h**
197 int **response_sd**
198 int **response_rc**
199 ros::Timer **myTimer**
200 int **dest**

Detailed Description

A class that sends/receives messages to/from the robot, and also publishes/subscribes to topics on ROS.

Constructor & Destructor Documentation

ROStoJAUSBridge::ROStoJAUSBridge ()

Constructor for **ROStoJAUSBridge**. Creates, binds to, and sends information along UDP socket upon construction call. Also sends a request_component_control message to the robot.

ROStoJAUSBridge::~ROStoJAUSBridge ()

Destructor for **ROStoJAUSBridge**. Closes UDP sockets for sending and receiving information.

Member Function Documentation

void ROStoJAUSBridge::CmdVelCallback (const geometry_msgs::TwistConstPtr & *msg*)

Sends velocity commands to robot. sends the robot velocity commands that were published on ROS.

void ROStoJAUSBridge::parse_msgs (const unsigned char * *buffer*, size_t *x*)

Parses messages sent back from the robot. Low priority messages are skipped.

void ROStoJAUSBridge::timerCallback (const ros::TimerEvent & *te*)

Initializes the buffers for sending messages to robot as well as receiving messages from robot. **timerCallback()** sends a query_platform_operational_data message and also parses messages sent back from the robot.

The documentation for this class was generated from the following files:

201 ROStoJAUSBridge.h

202 ROStoJAUSBridge.cpp

selected_camera_msg Class Reference

Inheritance diagram for selected_camera_msg:

Public Member Functions

203 virtual size_t **size** () const
204 virtual unsigned char * **marshal** (unsigned char *out_array) const
205 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
206 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * selected_camera_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t selected_camera_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * selected_camera_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

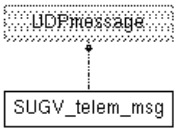
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

207 JAUSmessages.h
208 JAUSmessages.cpp

SUGV_telem_msg Class Reference

Inheritance diagram for SUGV_telem_msg:



Public Member Functions

209 virtual size_t **size** () const
210 virtual unsigned char * **marshal** (unsigned char *out_array) const
211 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
212 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * SUGV_telem_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t SUGV_telem_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * SUGV_telem_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

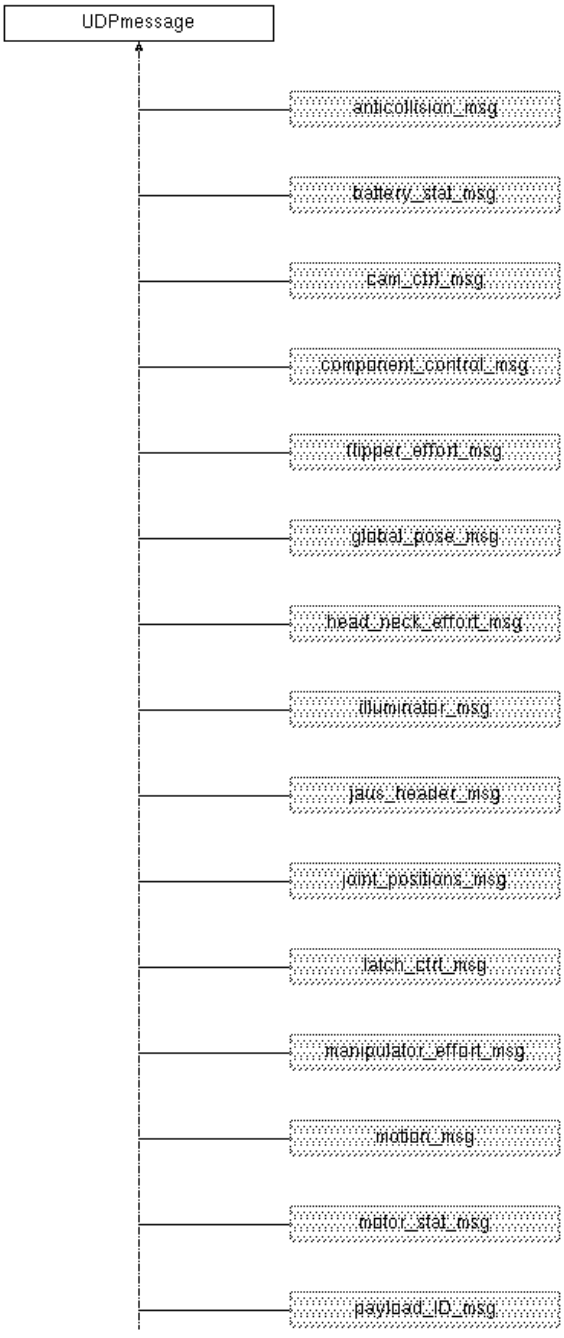
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

213 JAUSmessages.h
214 JAUSmessages.cpp

UDPmessage Class Reference

`#include <JAUSmessages.h>`
Inheritance diagram for UDPmessage:


UDPmessage
anticollision_msg
battery_stat_msg
cam_ctrl_msg
component_control_msg
flipper_effort_msg
global_pose_msg
head_neck_effort_msg
illuminator_msg
jaus_header_msg
joint_positions_msg
latch_ctrl_msg
manipulator_effort_msg
motion_msg
motor_stat_msg
payload_ID_msg

30

Public Member Functions

215 **UDPmessage ()**
216 virtual size_t **size ()** const =0
217 virtual unsigned char * **marshal (**unsigned char *outp_array**)** const =0
218 virtual const unsigned char * **unmarshal (**const unsigned char *inp_array**)**=0

Related Functions

(Note that these are not member functions.)

219 unsigned char * **WriteUChar (**unsigned char *buff, unsigned char val**)**
220 unsigned char * **WriteUShort (**unsigned char *buff, unsigned short val**)**
221 unsigned char * **IncludeTransVersion (**unsigned char *buff**)**
222 size_t **TransVersionSize ()**
223 float **TimeStampToSeconds (**unsigned int time**)**
224 const unsigned char * **ReadUChar (**const unsigned char *buff, unsigned char &val**)**
225 short **scaleToInt16 (**float val, float low, float high**)**
226 float **unscaleFromInt16 (**short scaledVal, float low, float high**)**
227 float **unscaleFromInt32 (**int scaledVal, float low, float high**)**

Detailed Description

Parent Class for most JAUS messages. Defines data members to be defined by all JAUS messages, including a message marshaling and unmarshaling system. (x_1,y_1)

Constructor & Destructor Documentation

UDPmessage::UDPmessage () [inline]

Constructor for **UDPmessage**.

Member Function Documentation

virtual unsigned char* UDPmessage::marshal (unsigned char * *outp_array*) const [pure virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implemented in **jaus_header_msg** (*p.13*), **component_control_msg** (*p.8*), **query_wrench_effort_msg** (*p.25*), **query_platform_op_data_msg** (*p.23*), **head_neck_effort_msg** (*p.11*), **flipper_effort_msg** (*p.9*), **wrench_effort_msg** (*p.35*), **manipulator_effort_msg** (*p.17*), **motion_msg** (*p.18*), **joint_positions_msg** (*p.14*), **global_pose_msg** (*p.10*), **velocity_msg** (*p.33*), **cam_ctrl_msg** (*p.8*), **anticollision_msg** (*p.6*), **motor_stat_msg** (*p.19*), **battery_stat_msg** (*p.6*), **platform_op_data_msg** (*p.21*), **illuminator_msg** (*p.12*), **power_ctrl_msg** (*p.21*), **selected_camera_msg** (*p.28*), **latch_ctrl_msg** (*p.14*), **SUGV_telem_msg** (*p.29*), **report_calibrate_device_msg** (*p.26*), **version_info_msg** (*p.34*), **payload_ID_msg** (*p.20*), **query_velocity_msg** (*p.24*), and **query_global_pose_msg** (*p.22*).

virtual size_t UDPmessage::size () const [pure virtual]

Returns size of message.

Implemented in **jaus_header_msg** (*p.13*), **component_control_msg** (*p.8*),

query_wrench_effort_msg (*p.25*), **query_platform_op_data_msg** (*p.23*), **head_neck_effort_msg** (*p.11*), **flipper_effort_msg** (*p.9*), **wrench_effort_msg** (*p.35*), **manipulator_effort_msg** (*p.17*), **motion_msg** (*p.18*), **joint_positions_msg** (*p.14*), **global_pose_msg** (*p.10*), **velocity_msg** (*p.33*), **cam_ctrl_msg** (*p.8*), **anticollision_msg** (*p.6*), **motor_stat_msg** (*p.19*), **battery_stat_msg** (*p.7*), **platform_op_data_msg** (*p.21*), **illuminator_msg** (*p.12*), **power_ctrl_msg** (*p.22*), **selected_camera_msg** (*p.28*), **latch_ctrl_msg** (*p.14*), **SUGV_telem_msg** (*p.29*), **report_calibrate_device_msg** (*p.26*), **version_info_msg** (*p.35*), **payload_ID_msg** (*p.20*), **query_velocity_msg** (*p.24*), and **query_global_pose_msg** (*p.22*).

virtual const unsigned char* UDPmessage::unmarshal (const unsigned char * *inp_array*) [pure virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implemented in **jaus_header_msg** (*p.13*), **component_control_msg** (*p.9*), **query_wrench_effort_msg** (*p.25*), **query_platform_op_data_msg** (*p.23*), **head_neck_effort_msg** (*p.11*), **flipper_effort_msg** (*p.9*), **wrench_effort_msg** (*p.35*), **manipulator_effort_msg** (*p.17*), **motion_msg** (*p.18*), **joint_positions_msg** (*p.14*), **global_pose_msg** (*p.10*), **velocity_msg** (*p.34*), **cam_ctrl_msg** (*p.8*), **anticollision_msg** (*p.6*), **motor_stat_msg** (*p.19*), **battery_stat_msg** (*p.7*), **platform_op_data_msg** (*p.21*), **illuminator_msg** (*p.12*), **power_ctrl_msg** (*p.22*), **selected_camera_msg** (*p.28*), **latch_ctrl_msg** (*p.15*), **SUGV_telem_msg** (*p.29*), **report_calibrate_device_msg** (*p.26*), **version_info_msg** (*p.35*), **payload_ID_msg** (*p.20*), **query_velocity_msg** (*p.24*), and **query_global_pose_msg** (*p.22*).

Friends And Related Function Documentation

unsigned char * IncludeTransVersion (unsigned char * *buff*) [related]

Writes the UDP transport version into array position pointed by buff. buff pointer is then incremented.

const unsigned char * ReadUChar (const unsigned char * *buff*, unsigned char & *val*) [related]

buff points to unsigned character value, which is stored in val. buff pointer is then incremented.

short scaleToInt16 (float *val*, float *low*, float *high*) [related]

Scales signed short value val, which is bounded by low and high. Shifts the center point of low and high to zero, and shifts val accordingly. Val is then upscaled by the ratio of the range of short values to the range of values from high to low.

float TimeStampToSeconds (unsigned int *time*) [related]

returns the value of JAUS timestamps in seconds.

size_t TransVersionSize () [related]

returns the size of the transport version character, which is 1 byte.

float unscaleFromInt16 (short *scaledVal*, float *low*, float *high*) [related]

Unscales a scaled short value scaledVal to lie between the bounds low and high.

float unscaleFromInt32 (int *scaledVal*, float *low*, float *high*) [related]

Unscales a scaled int value scaledVal to lie between the bounds low and high.

unsigned char * WriteUChar (unsigned char * *buff*, unsigned char *val*) [related]

Writes character val into array position pointed by buff. buff pointer is then incremented

unsigned char * WriteUShort (unsigned char * *buff*, unsigned short *val*) [related]

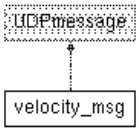
Writes unsigned short val into array positions pointed by buff. buff pointer is incremented by two bytes.

The documentation for this class was generated from the following file:

228 JAUSmessages.h

velocity_msg Class Reference

Inheritance diagram for velocity_msg:



Public Member Functions

229 int **getVel_x** ()
230 short **getYaw_rate** ()
231 unsigned int **getTime_stamp** ()
232 double **getSec** ()
233 virtual size_t **size** () const
234 virtual unsigned char * **marshal** (unsigned char *out_array) const
235 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
236 const unsigned char * **unmarshal** (const unsigned char *inp_array, const **jaus_header_msg** &head)

Member Function Documentation

unsigned char * velocity_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t velocity_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * velocity_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

237 JAUSmessages.h
238 JAUSmessages.cpp

version_info Struct Reference

Public Member Functions

239 **version_info** (unsigned char ID, unsigned char length, char *info)
240 **version_info** (const **version_info** &info)
241 **version_info** & **operator**= (const **version_info** &info)

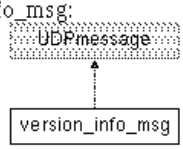
Public Attributes

242 unsigned char **device_ID**
243 unsigned char **length_version_string**
244 char * **version_string**

The documentation for this struct was generated from the following file:
245 JAUSmessages.h

version_info_msg Class Reference

Inheritance diagram for version_info_msg:



Public Member Functions

246 virtual size_t **size** () const
247 virtual unsigned char * **marshal** (unsigned char *out_array) const
248 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)
249 const unsigned char * **unmarshal** (const unsigned char *inp_array, **jaus_header_msg** &head)

Member Function Documentation

unsigned char * version_info_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t version_info_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * version_info_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

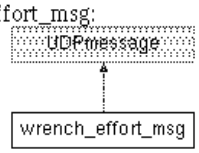
Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

250 JAUSmessages.h
251 JAUSmessages.cpp

wrench_effort_msg Class Reference

Inheritance diagram for wrench_effort_msg:



Public Member Functions

252 **wrench_effort_msg** (float linx, float angz)
253 virtual size_t **size** () const
254 virtual unsigned char * **marshal** (unsigned char *outp_array) const
255 virtual const unsigned char * **unmarshal** (const unsigned char *inp_array)

Member Function Documentation

unsigned char * wrench_effort_msg::marshal (unsigned char * *outp_array*) const [virtual]

Marshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.31*).

size_t wrench_effort_msg::size () const [virtual]

Returns size of message.

Implements **UDPmessage** (*p.31*).

const unsigned char * wrench_effort_msg::unmarshal (const unsigned char * *inp_array*) [virtual]

Unmarshals information specific to the JAUS message. Returns updated pointer.

Implements **UDPmessage** (*p.32*).

The documentation for this class was generated from the following files:

256 JAUSmessages.h
257 JAUSmessages.cpp

Index

INDEX

List of Symbols, Abbreviations, and Acronyms

ARL	U.S. Army Research Laboratory
DWA	Dynamic Window Approach
IMU	Inertial Measurement Unit
IP	Internet Protocol
JAUS	Joint Architecture for Unmanned Systems
Ladar	Laser Radar
OCU	Operator Control Unit
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SUGV	small unmanned ground vehicle
UDP	User Datagram Protocol

NO. OF COPIES	ORGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	US ARMY RSRCH DEV AND ENGRG CMND ARMAMENT RSRCH DEV & ENGRG CTR ARMAMENT ENGRG & TECHNLGY CTR ATTN AMSRD AAR AEF T J MATTS BLDG 305 ABERDEEN PROVING GROUND MD 21005-5001
1	US ARMY INFO SYS ENGRG CMND ATTN AMSEL IE TD A RIVERA FT HUACHUCA AZ 85613-5300
1	US GOVERNMENT PRINT OFF DEPOSITORY RECEIVING SECTION ATTN MAIL STOP IDAD J TATE 732 NORTH CAPITOL ST NW WASHINGTON DC 20402
2	US ARMY RDECOM TARDEC GROUND VEHICLE ROBOTICS ATTN AMSRD TAR R P FREDERICK ATTN AMSRD TAR S R KANIA 6501 E. ELEVEN MILE ROAD RDTA-RS WARREN, MI 48397
11	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CII A C RAO ATTN RDRL CIIA J ROGERS ATTN RDRL CII A H NGUYEN ATTN RDRL CII B L SADLER ATTN RDRL CII B L TOKARCIK ATTN RDRL CII B R WINKLER ATTN RDRL CII A S YOUNG ATTN RDRL CII A E STUMP ATTN RDRL CIO LL TECHL LIB ATTN RDRL CIO LT TECHL PUB ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK.